**CLOUD SECURITY**

These are the 5 common domains of cloud security:

* Identity and Access Management
* Logging and Monitoring
* Infrastructure Security
* Data Protection
* Incident Response

All AWS cloud security should start with identity, the first thing you do in the cloud is create the user which the user has access to all the infrastructure in the cloud and deploy resources.

**IDENTITY AND ACCESS MANAGEMENT**

IAM 1: Do not use the root account.

* The root user has complete access to all AWS services and resources in the account.
* Don not use the root user for your everyday tasks, even the administrative ones.
* We recommended creating an AWS IAM user, even for Admin.

**STEP1:**

Stop using the root account and create yourself a new account for administrative purposes. Use the new AWS IAM administrative account and not the root account you created during the sign-up. By not using it, you will reduce the chances of attackers.

Secondly, it is important to use individual uses for all AWS users in your organisation. You can do this by creating unique IAM accounts for all individuals in your organisation.

**STEP 2: Enable MFA**

ENABLE MULTI-FACTOR AUTHENTICATION on all interactive web console logins. Hackers getting passwords by key loggers is one of the major factors here and the use of MFA is going to stop hackers from doing this. With MFA enabled, when a user sign-in to the AWS website, they would be prompted for their username and password which is the first factor of what they know as well as authentication code from their MFA device or second authentication code. These together are called multiple factor authentications providing pre-security for your AWS resources e.g. EC2, EBS, Route 53 and so on.

AWS supports a lot of form factors for MFA including virtual tokens which can run on any application that supports the OTP standard. Google authenticator is a good example.

**IAM 3: Restrict long standing Access keys**

* Programmatic access to AWS typically uses access keys.
* Developers accessing AWS from their development laptops
* EC2 instances accessing the wider AWS eco-system
* Solution: Assume Role to request temporary, time bound credentials.

When developers want to programmatically access the AWS they use the access tokens to sign API poles that authenticate them with the end service, this may be directly from their laptop or they may place these access keys into code on EC2 instances. These access keys are long standing, developers can use them for a long period of time. To get the temporarily tokens, you can use authentity federation to assume a role inside AWS. If you need a token on an EC2 instance, the best is using an IAM role allowing the server to make an AWS STS call and receive temporarily tokens that allow the instance to interact with one another AWS services and Amazon S3 is a good example.

**LOGGIN AND MONITORING**

**LOG 1: Enable AWS CloudTrail**

With CloudTrail, you can log and continuously monitor AWS activities. CloudTrail provides event history of your AWS account activities including actions taken through the AWS management console, AWS STKs, command-line tools and other AWS services. This event history simplifies security analysis, resource tracking and troubleshooting. CloudTrail is enabled for 7 days by default.

**LOG 2: Enable Amazon GuardDuty**

This continually protects you from all malicious activities in the AWS account. By just clicking, Amazon GuardDuty can analyse millions of activities going on in the AWS services. It uses machine learning to detect attackers doing malicious activities in your account. It is easy to use and cost effective.

**LOG 3: BUILD AUTOMATED NOTIFICATION**

Cloud altravent, SNS and Lambda to secure your app from CloudTrail and GuardDuty and drop them in a messaging platform such as Slack, Jira etc.

**Infrastructure Security**

**INFRASTRUCTURE 1: PATCH YOUR SYSTEMS**

AWS system manager is a tool that can help here. It helps by maintaining system compliance by scanning your system against your patch and system policies, you can update firewall policies and implement anti-virus policies here.

**INFRASTRUCTURE 2: REDUCE YOUR ATTACK SURFACES**

Amazon Virtual private cloud allows you to find both public and private subnet. Amazon cloudfront or Elastic Load balancers should not be set to a public network or the internet.

Secondly is to ensure your security group both public and private subnet are appropriate. Finally is to find a way to connect this systems to a VPN and not the internet

**INFRASTRUCTURE 3: CREATE A DDOS RESILIENT ARCHITECTURE**

You have to protect yourself from Distributed denial of service attacks. Route 53 helps to manage DNS services. You can use Amazon Route 53, Amazon CloudFront, AWS WAF and AWS Shield to protect against DDoS attacks.

**DATA PROTECTION**

**DATA PROTECTION 1: Encrypt Everything**

AWS Key Management Service to encrypt your data.

**DATA PROTECTION 2: IMPLEMENT STRONG ACCESS CONTROLS**

**PREVENT**

IAM policies

**Detect**

CloudWatch Events

Configuration Rules

SNS

AWS IAM helps to keep data secured in the S3 buckets and prevents them from being public in the first place.

Amazon CloudWatch Event helps to detect whether data in the S3 bucket has been set to public or not.

**DATA PROTECTION 3: DO NOT STORE SECRETS IN CLEAR-TEXT**

Helps to protect the services and data in the database on your AWS account. It also helps to protect secret data using audits.

**INCIDENT RESPONSE**

**INCIDENT RESPONSE 1: HAVE A PLAN AND TEST IT**

**ELEMENTS OF AN INCIDENT RESPONSE PLAN**

* Roles and Responsibilities
* Identity
* Contain
* Respond
* Recover

**INCIDENT RESPONSE 2: BACKUP AND RESTORE**

For EC2 instances and EBS you can use snapshot are backup devices for storing data. For keeping multiple versions of the same object in the S3 buckets, you can use Amazon Versioning to back up the data in the S3 buckets.

**Sources of Best practices**

* AWS Cloud Adoption Framework
* CIS Foundations Framework <http://bit.ly/aws-cis> (centre for internet security)
* AWS Well-Architected

**CONCLUSION**

Build a security backlog and implement security as features

Cover the 5 core epics which are

* Identity
* Logging and Monitoring
* Infrastructure Security
* Data Protection
* Incident Response

Use AWS native security features to help your security team journey.

**HOW TO TROUBLESHOOT TREATS ON THE CLOUD**

**USE CASE**

For instance, you received a random message on Facebook stating that you just won a million naira cash prize, to claim your cash prize, please click on this link to fill in your credentials. This is actually a Spam but the user did not know. Facebook knows this kind of threat through the following stages.

**STEP 1:**

The first step is monitoring data so you have ai algorithms which knows what system behaviour is and any deviation from this normal system behaviour creates an alarm and this alarm is then monitored by the cloud experts or the cloud security experts sitting over there and there is a threat which is why they go into the next step.

**STEP 2**

Gaining visibility so you should understand what caused that problem or who caused that problem so your cloud security experts look for tools which give them the ability to look into the data and find the problem which caused the problem.

**STAGE 3**

Managing access gives you a list of users who have access pinpoint the user who did that and that user will be wiped out of the system using the managing access stage.

**HOW TO IMPLEMENT THESE STAGES IN AWS**

If you have an app in AWS experiencing this same problem, there is a service in AWS called the CloudWatch which is a cloud monitoring tool, it can monitor your EC2 and your other AWS resources on CloudWatch, you can monitor the network traffic in the AWS resources and create alarms in your CloudWatch so if there is a deviation from normal system behaviour. To access it, you go to management tools and click on CloudWatch, then click on metrix and you will see a list where you can either monitor your EC2, EBS or S3. For instance, if you want to monitor your EC2, then click on network out to see a graph of the discrete ops. You can also set alarms there. Click on create alarms, then click on EC2 and you can select the metrix from there, click on next to set alarm name and description you want to give it. Then go to actions to see if when an alarm is triggered, what actions should perform, if the alarm rings, you can send a notification to your SNS topic. SNS is a protocol that allows you to set who you want to view the notification. Notify me is a topic in SNS in which you put in your email address there.

**GAINING VISIBILITY**

There is a service in AWS called the CloudTrail for gaining visibility. This is basically a logging service where each and every log to each and every API call is made. On the AWS dashboard, you will go to CloudTrail services in the management tools and you can see all the logs and set the time frame from 2pm till when the hacker gained access to the system.

**MANAGING ACCESS**

By using AWS IAM, you can wipe out the hacker from your system and clear the logs. AWS IAM basically authenticates that service. You have a root user or when you sign-up for the first time you open a root account. You have employees and you want to give specific or granular permissions to a particular employee since he is capable of tracking hackers. AWS IAM gives the person a private file which is free to use. On the AWS dashboard, go to the security and compliance domain and click on IAM. Click on roles, then click on the EC2 instance role or any role that created the threat or problem. Go to revoke sessions and click on revoke access sessions and hence I will be able to wipe out that user from accessing my AWS resources. To give a particular employee free access to confidential files, you go to roles and click on create new role, then put in a name say HELLO, then click on next step and go to role for ing provider access and you can select how that user will be accessing your AWS account. Then click on Facebook and give it some random application Id, then click on next to get the policy document so click on next step and here you attach a policy or permissions you want to grant that user.

Write a python program to count the number of capital letters in a text. The user will enter the text himself.